

# **MOLLUSCAN SHELL DATABASE DESIGN**

Submitted in partial fulfilment of the requirement for the award of  
Bachelor of Engineering Degree in  
Electronics and Communication Engineering

by

**ADITYA KAUSHAL (40130002)**

**ROHIT SHARMA (40130198)**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING  
SCHOOL OF ELECTRICAL AND ELECTRONICS**

## **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**

**CATEGORY-1 UNIVERSITY BY UGC**

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by  
AICTE**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

**APRIL – 2024**



**SATHYABAMA**

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **Aditya Kaushal (40130002), Rohit Sharma (40130198)** who carried out the project entitled **MOLLUSCAN SHELL DATABASE DESIGN** under my supervision from **DECEMBER 2023 to APRIL 2024**.

  
Internal Guide

**Dr. G. RAJALAKSHMI, M.E., Ph.D.**

  
Head of the Department

**Dr. T. RAVI, M.E., Ph.D.**

---

Submitted for Viva voce Examination held on \_\_\_\_\_

Internal Examiner

External Examiner

## DECLARATION

We **ADITYA KAUSHAL (40130002)** and **ROHIT SHARMA (40130198)** hereby declare that the Project Report entitled **"MOLLUSCAN SHELL DATABASE DESIGN"** done by us under the guidance of **Dr. G. RAJALAKSHMI, M.E, Ph.D.** at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **ELECTRONICS AND COMMUNICATION ENGINEERING**.

DATE: 18<sup>th</sup> April 2024

PLACE: Chennai

### SIGNATURE OF THE CANDIDATES

1. Aditya Kaushal
2. Rohit Sharma

## **ACKNOWLEDGEMENT**

We are pleased to acknowledge our sincere thanks to the **Board of Management** of **SATHYABAMA INSTITUTE of SCIENCE and TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D., Dean, School of Electrical and Electronics Engineering** and **Dr. T. RAVI, M.E., Ph.D. Head of the Department, Department of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. G. RAJALAKSHMI, M.E., Ph.D. Dept. of Electronics and Communication Engineering** for her valuable guidance, suggestions and constant encouragement for the successful completion of our project work.

We wish to express our thanks to all teaching and non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

## **ABSTRACT**

The Molluscan Shell Database Design aims to create a comprehensive and organized system for storing and managing data related to molluscan shells. Molluscan shells are crucial for various scientific studies, including biodiversity assessments, ecological research, and evolutionary biology. The database design will incorporate efficient data structures and relational models to accommodate diverse information. Utilizing modern database technologies, the system will ensure scalability, data integrity, and ease of access for researchers and enthusiasts. By structuring the data effectively, the database design facilitates seamless querying, analysis, and visualization of molluscan shell information, thereby contributing to advancements in marine biology and conservation efforts.

## TABLE OF CONTENTS

| CHAPTER NO. | TITLE                                          | PAGE NO. |
|-------------|------------------------------------------------|----------|
|             | ABSTRACT                                       | v        |
|             | LIST OF ABBREVIATIONS                          | viii     |
|             | LIST OF FIGURES                                | ix       |
| 1           | INTRODUCTION                                   | 1        |
|             | 1.1 Problem Statement                          | 1        |
|             | 1.2 Objective of the Project                   | 2        |
|             | 1.3 Scope                                      | 3        |
|             | 1.4 Project Introduction                       | 3        |
| 2.          | LITERATURE SURVEY                              | 4        |
|             | 2.1 Related Work                               | 4        |
| 3.          | AIM AND SCOPE OF PRESENT INVESTIGATION         | 6        |
|             | 3.1 Existing System                            | 6        |
|             | 3.2 Disadvantages                              | 6        |
|             | 3.3 Proposed System                            | 7        |
|             | 3.4 Advantages                                 | 7        |
|             | 3.5 Work Flow of Proposed system               | 8        |
|             | 3.6 Functional and Non Functional Requirements | 8        |
|             | 3.7 Software Requirements                      | 10       |
|             | 3.8 Hardware Requirements                      | 10       |

|    |                                    |    |
|----|------------------------------------|----|
| 4. | <b>METHODS AND ALGORITHMS USED</b> | 11 |
|    | 4.1 Introduction of Input design   | 11 |
|    | 4.2 UML Diagram                    | 12 |
|    | 4.3 Data Flow Diagram              | 19 |
| 5. | <b>RESULTS AND DISSCUSION</b>      | 21 |
|    | 5.1 Modules                        | 21 |
|    | 5.2 Feasible Study                 | 22 |
|    | 5.3 Types of test & Test Cases     | 26 |
| 6. | <b>CONCLUSION</b>                  | 30 |
|    | <b>REFERENCE</b>                   | 31 |
|    | <b>APPENDIX A: STANDARDS</b>       | 33 |
|    | <b>APPENDIX B: CONSTRAINTS</b>     | 33 |
|    | <b>APPENDIX C: TRADE OFFS</b>      | 33 |

## LIST OF ABBREVIATIONS

| Abbreviation | Definition                                   | Page no. |
|--------------|----------------------------------------------|----------|
| IDE          | Integrated Development Environment           | 10       |
| RAM          | Random Access Memory                         | 10       |
| GB           | Giga Byte                                    | 10       |
| UML          | Unified Modelling Language                   | 12       |
| DFD          | Data Flow Diagram                            | 19       |
| HTML         | Hyper Text Markup Language                   | 33       |
| CSS          | Cascading Style Sheets                       | 33       |
| ECMA         | European Computer Manufacturer's Association | 33       |

## LIST OF FIGURES

| <b>Fig. No.</b> | <b>Description</b>          | <b>Page no.</b> |
|-----------------|-----------------------------|-----------------|
| 3.1             | Work Flow                   | 8               |
| 4.1             | Use Case Diagram            | 13              |
| 4.2             | Class Diagram               | 14              |
| 4.3             | Sequence Diagram            | 15              |
| 4.4             | Collaboration Diagram       | 16              |
| 4.5             | Deployment Diagram          | 17              |
| 4.6             | Activity Diagram            | 18              |
| 4.7             | Component Diagram           | 19              |
| 4.8             | Level Diagram 1             | 20              |
| 4.9             | Level Diagram 2             | 20              |
| 5.1             | Sign Up                     | 21              |
| 5.2             | Sign In                     | 22              |
| 5.3             | Front Page                  | 23              |
| 5.4             | Contact Page                | 24              |
| 5.5             | Gallery of Molluscan Shells | 25              |
| 5.6             | Sample Collection           | 28              |

# **CHAPTER-1**

## **INTRODUCTION**

### **1.1 PROBLEM STATEMENT**

Despite the significant role molluscan shells play in various scientific endeavors such as biodiversity assessments, ecological research, and evolutionary biology, there is a notable lack of a centralized and comprehensive database specifically designed to store and manage data related to molluscan shells. This absence hinders efficient data retrieval, analysis, and collaboration among researchers and enthusiasts in the field. Existing databases often lack the necessary structure and organization to handle the diverse range of information associated with molluscan shells, leading to inefficiencies and limitations in data management. Additionally, with the increasing volume and complexity of data generated in this field, there is a growing need for a scalable and robust database solution that can accommodate future expansions and advancements in research. Therefore, developing a dedicated Molluscan Shell Database Design is essential to address these challenges and facilitate seamless access, exploration, and utilization of molluscan shell data for scientific inquiry and conservation efforts.

### **1.2 MOTIVATION**

The Molluscan Shell Database Design project aims to streamline access to crucial molluscan shell data for scientific research. Existing methods often lack organization and accessibility, hindering biodiversity assessments and ecological studies. By implementing efficient data structures and relational models, this project creates a scalable and organized system. Utilizing modern database technologies ensures data integrity and ease of access. Researchers can seamlessly query, analyze, and visualize molluscan shell information, facilitating advancements in marine biology and conservation efforts.

### **1.3 OBJECTIVE OF THE PROJECT**

The objective of the Molluscan Shell Database Design project is to develop a comprehensive and efficient system for storing, managing, and accessing data related to molluscan shells. The project aims to achieve the following specific objectives.

- **Data Organization:** Create a structured and organized database schema to accommodate diverse types of information associated with molluscan shells, including taxonomy, morphology, distribution, habitat, ecological interactions, and conservation status.
- **Data Integration:** Integrate existing datasets and sources of molluscan shell data into the database to ensure completeness and maximize the utility of the database for researchers and enthusiasts.
- **Scalability:** Design the database architecture to be scalable, capable of handling large volumes of data, and adaptable to future expansions and advancements in research.
- **Data Integrity:** Implement measures to ensure data accuracy, consistency, and reliability, such as data validation rules, referential integrity constraints, and standardized data entry protocols.
- **User Interface:** Develop user-friendly interfaces for data input, querying, analysis, and visualization to facilitate ease of use and accessibility for researchers and enthusiasts with varying levels of technical expertise.
- **Interdisciplinary Collaboration:** Promote interdisciplinary collaboration by providing mechanisms for data sharing, collaboration, and integration with other research databases and platforms in related fields such as marine biology, ecology, and biodiversity conservation.
- **Advancements in Research:** Enable researchers to conduct advanced data analysis, comparative studies, and visualization techniques to gain insights into molluscan diversity, distribution patterns, ecological interactions, evolutionary processes, and conservation priorities.

- **Educational Resources:** Provide educational resources and tools within the database to support learning, teaching, and outreach activities related to molluscan biology, ecology, and conservation.

By achieving these objectives, the project aims to contribute to the advancement of scientific knowledge, facilitate evidence-based decision-making in conservation efforts, and promote greater understanding and appreciation of molluscan diversity and ecology.

#### **1.4 SCOPE:**

The project scope encompasses collecting comprehensive molluscan shell data, designing a scalable and secure database, developing user-friendly interfaces for data access and analysis, ensuring data integrity and interoperability with other systems, providing thorough documentation and training materials for users, and incorporating iterative enhancements based on user feedback. It aims to create a robust and accessible platform that facilitates scientific research, education, and conservation efforts related to molluscan shells, ultimately advancing understanding of their biodiversity, distribution, and ecological significance while supporting evidence-based decision-making and collaboration across interdisciplinary fields.

#### **1.5 PROJECT INTRODUCTION**

The Molluscan Shell Database Design project introduces a structured approach to managing and accessing data related to molluscan shells, vital for various scientific studies. Existing methods often lack organization and accessibility, hindering biodiversity assessments, ecological research, and evolutionary biology. This project aims to address these challenges by creating a comprehensive and organized database. Utilizing efficient data structures and relational models, the system ensures scalability, data integrity, and ease of access. By facilitating seamless querying, analysis, and visualization of molluscan shell information, the project contributes significantly to advancements in marine biology and conservation efforts.

## **CHAPTER-2**

### **LITERATURE SURVEY**

#### **2.1 RELATED WORK**

##### **The shell structure and mineralogy of the bivalvia.**

Understanding the shell structure and mineralogy of bivalve mollusks is fundamental to comprehending their biology, ecology, and evolution. Taylor, Kennedy, and Hall's comprehensive work provides insights into the intricate architecture and mineral composition of bivalve shells, shedding light on their mechanical properties, growth patterns, and environmental adaptations.

Taylor, Kennedy, and Hall's seminal work extensively examines the shell structure and mineralogy of bivalve mollusks. Through detailed investigations, they elucidate the complex arrangement of organic and inorganic components within the shell, highlighting its role in protecting the soft tissues, regulating buoyancy, and facilitating locomotion. By synthesizing existing knowledge and presenting new findings, this reference serves as a cornerstone for understanding the morphological diversity and ecological significance of bivalve shells.

##### **A new critical estimate of named species-level diversity of the recent Mollusca.**

Accurate estimation of species-level diversity is essential for biodiversity assessments, conservation planning, and ecological research. Rosenberg's study provides a comprehensive evaluation of the named species-level diversity within the Mollusca, offering valuable insights into the taxonomic richness and distribution patterns of this diverse phylum. Rosenberg's critical assessment of named species-level diversity within the Mollusca highlights the challenges and uncertainties associated with taxonomic classification and species delimitation. By synthesizing data from multiple sources and employing rigorous statistical methods, he provides a revised estimate of molluscan diversity, emphasizing the need for continued taxonomic research and data integration. This reference serves as a valuable

resource for understanding the global distribution and evolutionary history of mollusks.

**Speciation and diversity on tropical rocky shores: a global phylogeny of snails of the genus *Echinolittorina*.**

Tropical rocky shores harbor diverse assemblages of gastropod mollusks, offering unique opportunities to study speciation, adaptation, and community dynamics. Williams and Reid's research focuses on the genus *Echinolittorina*, providing insights into the evolutionary processes driving diversification and ecological specialization in these snails. Williams and Reid's global phylogenetic analysis of *Echinolittorina* snails elucidates the patterns and mechanisms of speciation on tropical rocky shores. By integrating molecular data with ecological and morphological information, they reconstruct the evolutionary history of this genus, revealing cryptic diversity and phylogeographic patterns. Their findings underscore the role of habitat heterogeneity and geographic isolation in shaping snail diversity, with implications for marine conservation and management.

**Revised classification, nomenclator and typification of gastropod and monoplacophoran families.**

A comprehensive and standardized classification system is essential for organizing the vast diversity of gastropod and monoplacophoran mollusks. Bouchet et al.'s revision provides a comprehensive framework for classifying these taxa, clarifying their evolutionary relationships and facilitating communication among researchers and taxonomists. Bouchet et al.'s revised classification, nomenclator, and typification of gastropod and monoplacophoran families represent a significant milestone in molluscan taxonomy. Through meticulous examination of morphological and molecular data, they propose a systematic arrangement of families, subfamilies, and genera, reflecting the latest phylogenetic insights. This reference serves as an invaluable resource for researchers, educators, and conservationists involved in the study and conservation of gastropod and monoplacophoran biodiversity.

## **CHAPTER-3**

### **AIM AND SCOPE OF PRESENT INVESTIGATION**

#### **3.1 Existing System**

The existing system for managing molluscan shell data is fragmented, lacking a centralized database dedicated to this field. Data sources are often disparate, with limited standardization and accessibility. Researchers rely on manual methods for data storage, retrieval, and analysis, leading to inefficiencies and inconsistencies. Existing databases may not adequately address the diverse needs of researchers and lack features for comprehensive data management and analysis. Moreover, interoperability with other research platforms is limited, hindering collaboration and knowledge exchange. Overall, the current system lacks integration, standardization, and user-friendliness, underscoring the need for a dedicated molluscan shell database design to address these shortcomings.

#### **3.2 Disadvantages**

The disadvantages of the existing system for managing molluscan shell data include:

- **Fragmentation:** Data is scattered across multiple sources, making it challenging to access and integrate information effectively.
- **Lack of Centralization:** There is no centralized database dedicated to molluscan shell data, leading to inefficiencies in data storage, retrieval, and management.
- **Limited Standardization:** Data may lack standardized formats and terminology, leading to inconsistencies and difficulties in data interpretation and analysis.
- **Manual Processes:** Researchers rely on manual methods for data entry and analysis, which are time-consuming and prone to errors.

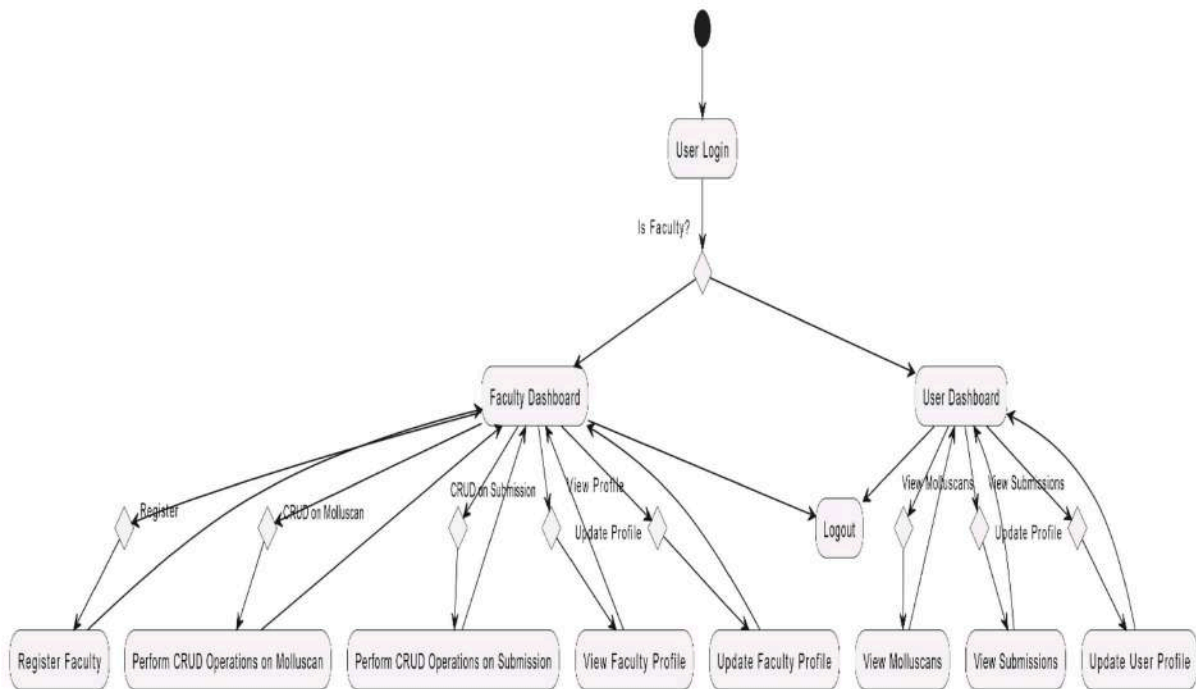
### **3.3 Proposed System**

The proposed method involves rigorous requirement analysis, relational database design, integration of diverse datasets, implementation using modern technologies, intuitive user interface development, validation for data integrity, interoperability with other platforms, comprehensive documentation, and training materials. Testing ensures functionality and usability, with deployment and ongoing maintenance for continuous improvement. This method aims to create a scalable, secure, and user-friendly molluscan shell database facilitating efficient data management, analysis, and collaboration. Ultimately, it supports scientific research, education, and conservation efforts, addressing existing limitations in data fragmentation, standardization, and accessibility within the field.

### **3.4 Advantages**

- **Centralized Data Management:** The database provides a centralized repository for molluscan shell data, enabling efficient storage, retrieval, and management of diverse information types, enhancing organization and accessibility for researchers and enthusiasts.
- **Enhanced Data Integrity:** Through validation rules and quality control measures, the database ensures data accuracy, consistency, and reliability, reducing errors and inconsistencies commonly associated with manual data entry and management processes.
- **Improved Collaboration and Interoperability:** Interoperability features facilitate seamless data sharing and collaboration with other research databases and platforms, fostering interdisciplinary research initiatives and knowledge exchange within the scientific community.
- **User-Friendly Interface:** Intuitive user interfaces make data input, querying, analysis, and visualization accessible to users with varying levels of technical expertise, promoting usability and adoption among researchers, educators, and conservationists.

### 3.5 Work Flow of Proposed System



**Fig:3.1: Work Flow**

### 3.6 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

#### 3.6.1 Functional Requirements

- **Data Entry and Management:** The system should allow users to input and manage data related to molluscan shells, including taxonomy, morphology, distribution, and ecological traits. Users should be able to create, update, and delete shell records as needed.
- **Search and Querying:** The system should support advanced search functionalities, allowing users to query shell data based on various criteria such as species name, geographic location, habitat type, and shell characteristics. Queries should return relevant results efficiently, displaying detailed information about matching shell records.
- **Visualization and Reporting:** The system should provide visualization tools to

generate charts, graphs, and maps based on queried data, enabling users to analyze trends and patterns in molluscan shell distributions and characteristics. Users should be able to generate customizable reports summarizing queried data for further analysis or presentation purposes.

- **Taxonomic Classification:** The system should incorporate a standardized taxonomic classification scheme for molluscan shells, allowing users to navigate and explore hierarchical relationships among species, genera, families, and higher taxonomic ranks. Taxonomic information should be updated regularly to reflect the latest revisions and discoveries in molluscan taxonomy.
- **User Authentication and Access Control:** The system should support user authentication mechanisms to ensure secure access to sensitive data. Role-based access control should be implemented to restrict user privileges based on their roles and responsibilities within the organization.

### **3.6.2 Non-Functional Requirements**

- **Performance:** The system should exhibit fast response times, even when handling large volumes of data and concurrent user requests. Queries and data retrieval operations should be optimized for efficiency to minimize latency and maximize user productivity.
- **Scalability:** The system should be scalable to accommodate growing datasets and increasing user demand over time. It should be capable of scaling horizontally by adding additional hardware resources or vertically by optimizing database performance and architecture.
- **Reliability:** The system should be highly reliable, ensuring data integrity, consistency, and availability at all times. Measures such as data backup, replication, and disaster recovery should be implemented to mitigate the risk of data loss or corruption.
- **Usability:** The system should have an intuitive user interface with clear navigation and user-friendly features. Training and documentation should be provided to assist users in effectively utilizing the system's functionalities and

maximizing productivity.

- Security: The system should enforce robust security measures to protect sensitive data from unauthorized access, modification, or disclosure. Data encryption, secure authentication protocols, and audit trails should be implemented to safeguard against security threats and breaches.

### **3.7 SOFTWARE REQUIREMENTS**

- Operating System : Windows95/98/2000/XP
- Application Server : Tomcat 7.0
- Front End : React JS
- Scripts : JavaScript.
- Backend Language : Java
- Database : MySQL 6.0
- IDE : IntelliJ and Visual Studio Code

### **3.8 HARDWARE REQUIREMENTS**

- **Processor** : Intel i3
- RAM : 4GB
- Hard Disk : 500 GB

## **CHAPTER-4**

### **METHODS AND ALGORITHMS USED**

#### **4.1 Introduction of Input Design**

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties –

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding
  - What are the inputs needed for the system?
  - How end users respond to different elements of forms and screens.

#### **Objectives for Input Design**

The objectives of input design are –

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.

To use validation checks and develop effective input controls.

#### **Output Design**

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

### **Objectives of Output Design**

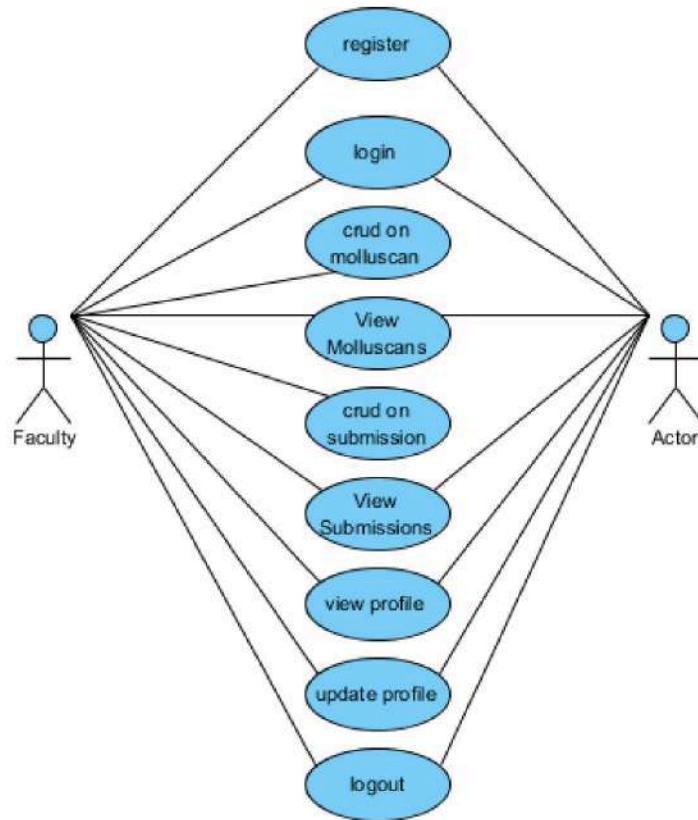
The objectives of input design are:

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user's requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

## **4.2 UML Diagrams**

### **4.2.1 Use Case Diagram**

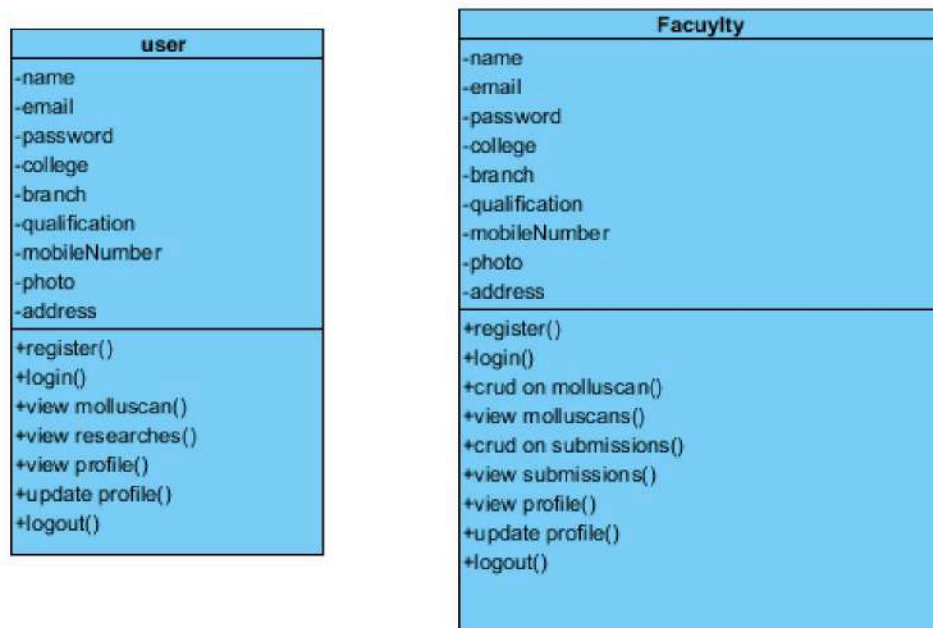
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. Actors are external entities that interact with the system. These can include users, other systems, or hardware devices. In the context of a Use Case Diagram, actors initiate use cases and receive the outcomes. Proper identification and understanding of actors are crucial for accurately modeling system behavior.



***Fig:4.1: Use Case Diagram***

#### **4.2.2 Class Diagram**

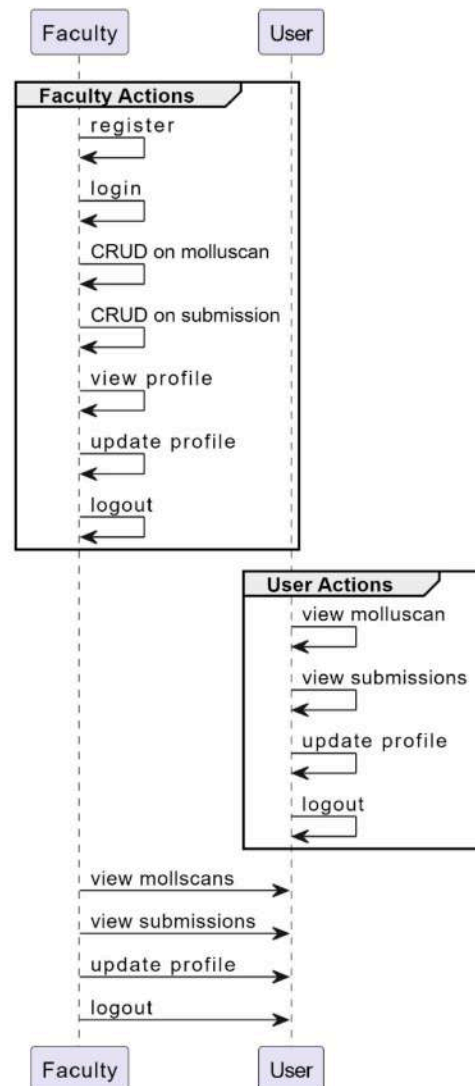
In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**Fig:4.2: Class Diagram**

#### 4.2.3 Sequence Diagram

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. It is a modeling language in the field of software engineering that aims to set standard ways to visualize the design of a system. UML guides the creation of multiple types of diagrams such as interaction, structure, and behavior diagrams. A sequence diagram is the most commonly used interaction diagram.

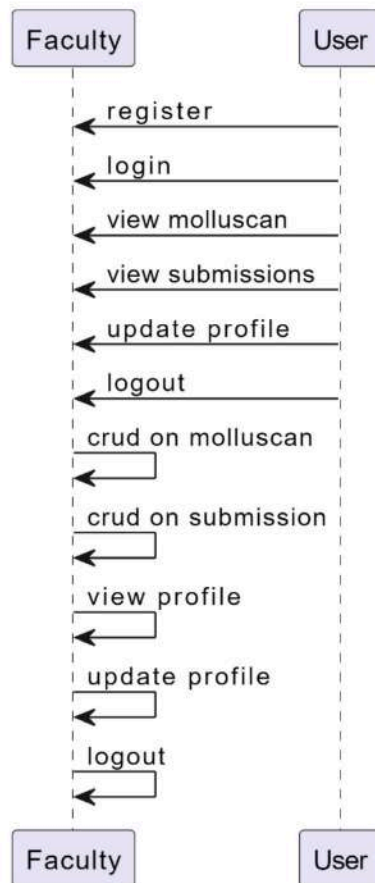


**Fig:4.3: Sequence Diagram**

#### 4.2.4 Collaboration Diagram

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. The same order of management system has been taken to describe the collaboration diagram. The method calls are similar to that of a sequence diagram.

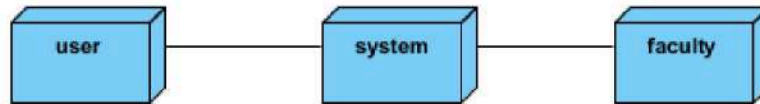
But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.



**Fig:4.4: Collaboration Diagram**

#### 4.2.5 Deployment Diagram

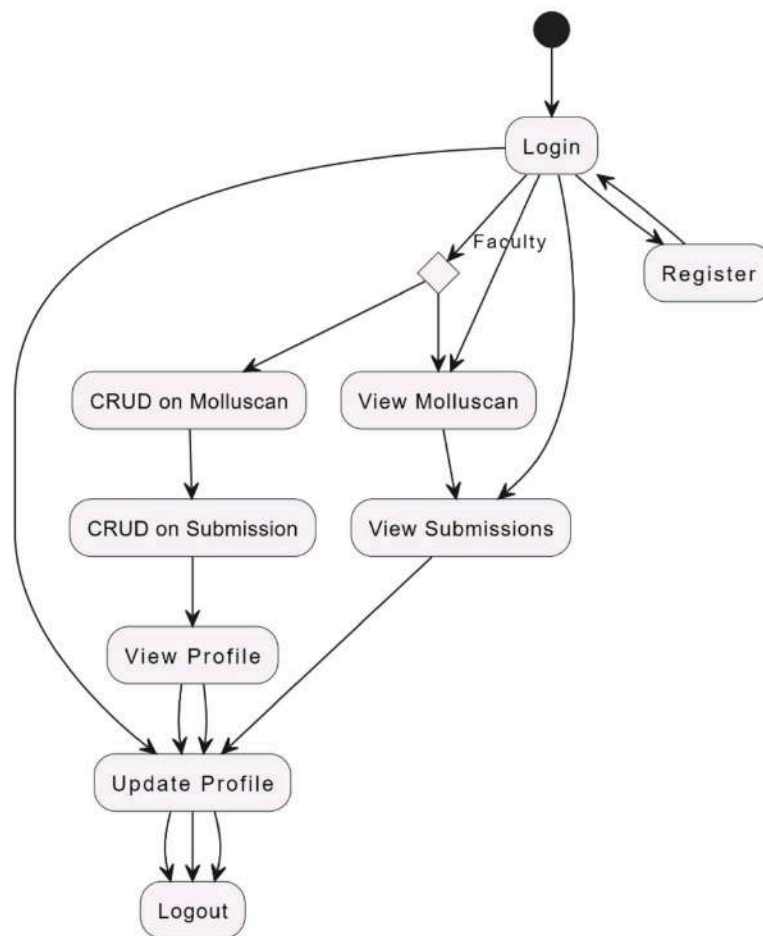
Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.



***Fig:4.5: Deployment Diagram***

#### **4.2.6 Activity Diagram**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control. Activity Diagrams are used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. It is a type of behavioral diagram and we can depict both sequential processing and concurrent processing of activities using an activity diagram i.e. an activity diagram focuses on the condition of flow and the sequence in which it happens. A process can have only one initial state unless we are depicting nested activities. We use a black filled circle to depict the initial state of a system. For objects, this is the state when they are instantiated. The Initial State from the UML Activity Diagram marks the entry point and the initial Activity State. An activity represents execution of an action on objects or by objects. We represent an activity using a rectangle with rounded corners. Basically any action or event that takes place is represented using an activity.

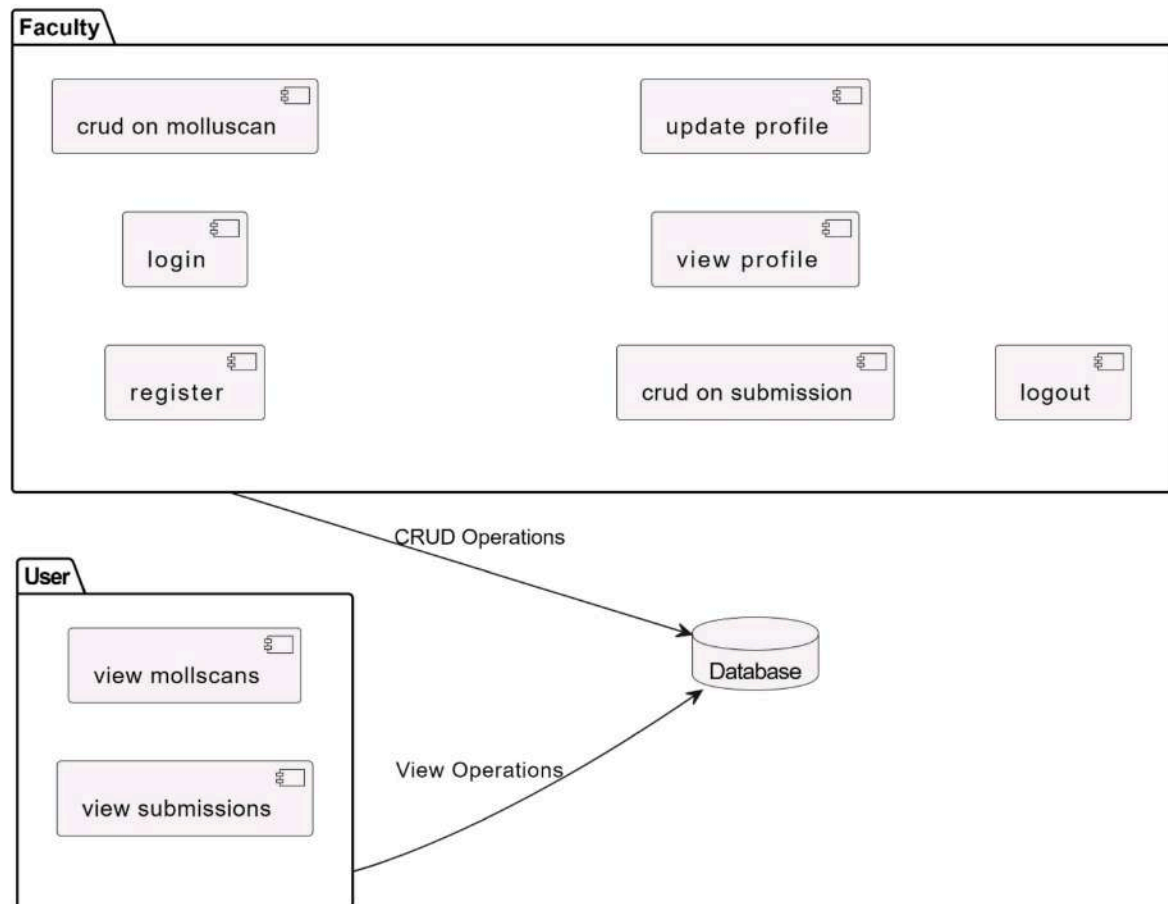


**Fig:4.6: Activity Diagram**

#### 4.2.7 Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development. A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node. It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A

component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained by the provided and required interfaces.



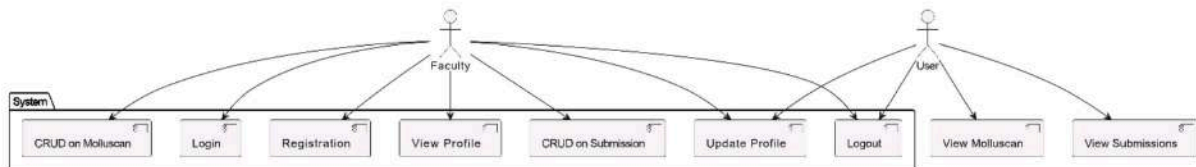
**Fig:4.7: Component Diagram**

### 4.3 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and

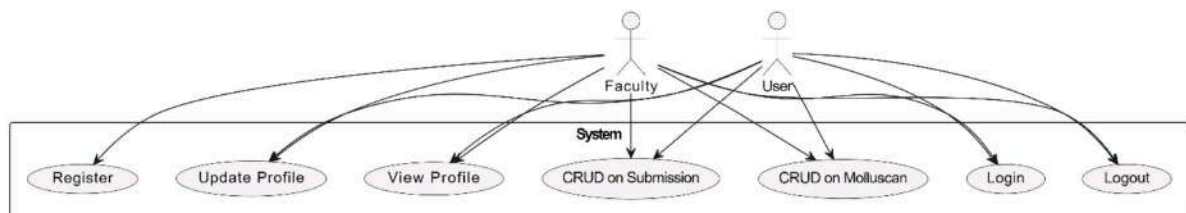
boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. It is a graphical tool, useful for communicating with users ,managers and other personnel. it is useful for analyzing existing as well as proposed system.

### Level 1 Diagram:



**Fig:4.8: Level Diagram 1**

### Level 2 Diagram:



**Fig:4.9: Level Diagram 2**

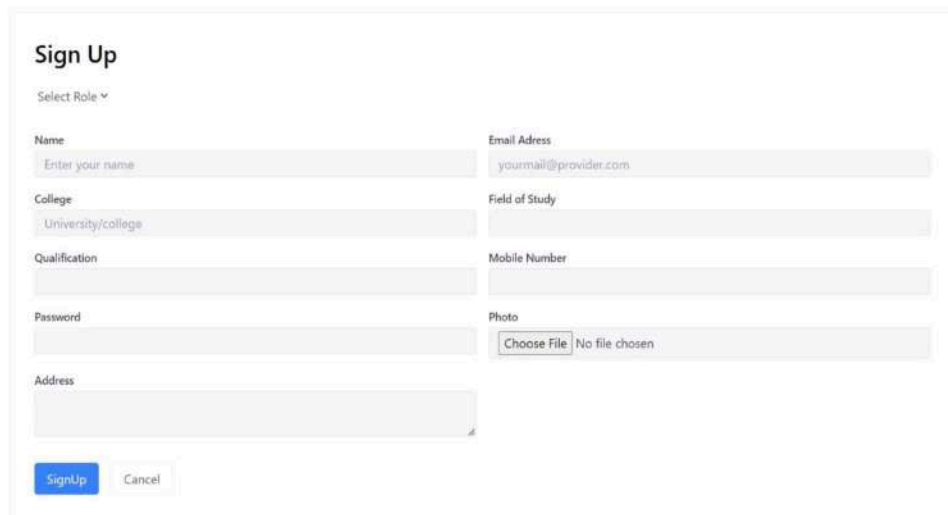
## CHAPTER-5

### RESULTS AND DISCUSSION

#### 5.1 Modules

➤ Faculty:

- Register: Faculty register into the application with his details
- Login: Faculty login to the application with valid credentials
- Add Molluscan: Faculty can add the Molluscan details
- View Molluscan: Faculty can see the molluscan details added by him
- Add Researches: Faculty can add researches
- View Researches: Faculty can view researches added by him
- Profile: Faculty can view and update their profile
- Logout: Faculty must logout



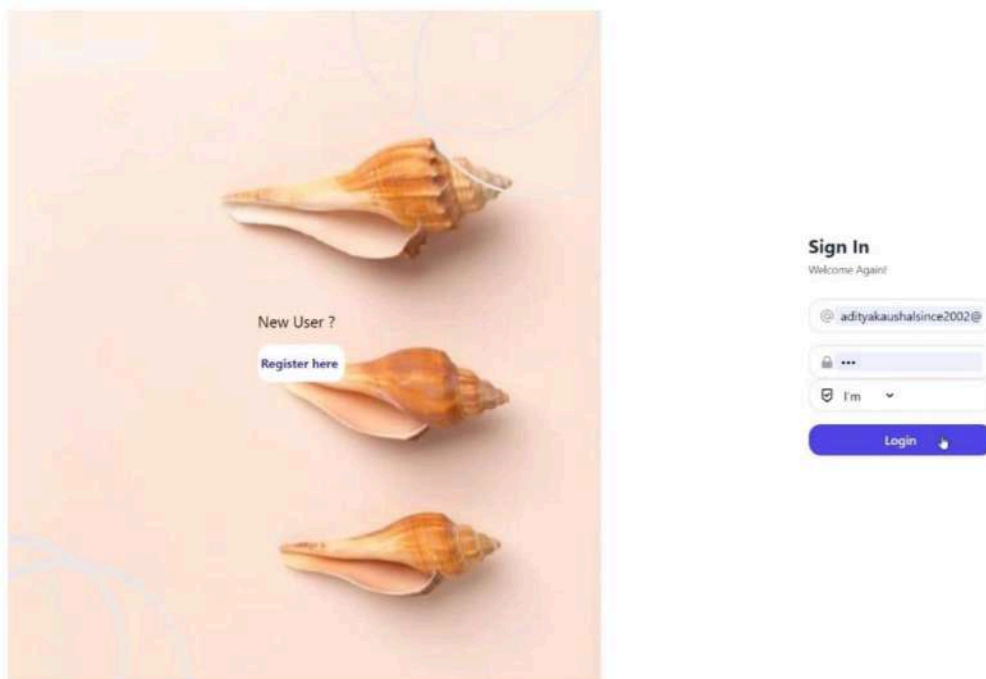
The image shows a 'Sign Up' form with the following fields and controls:

- Select Role**: A dropdown menu.
- Name**: Text input with placeholder 'Enter your name'.
- Email Address**: Text input with placeholder 'yourmail@provider.com'.
- College**: Text input with placeholder 'University/college'.
- Field of Study**: Text input.
- Qualification**: Text input.
- Mobile Number**: Text input.
- Password**: Text input.
- Photo**: File upload control with a 'Choose File' button and 'No file chosen' text.
- Address**: Text input.
- Buttons**: 'SignUp' (blue) and 'Cancel' (grey) buttons at the bottom left.

**Fig:5.1: Sign Up**

➤ Student:

- Register: Student register into the application with his details
- Login: Student login to the application with valid credentials
- View Molluscan: Student can see the molluscan details
- View Researches: Student can view researches
- Profile: Student can view and update their profile
- Logout: Student must logout



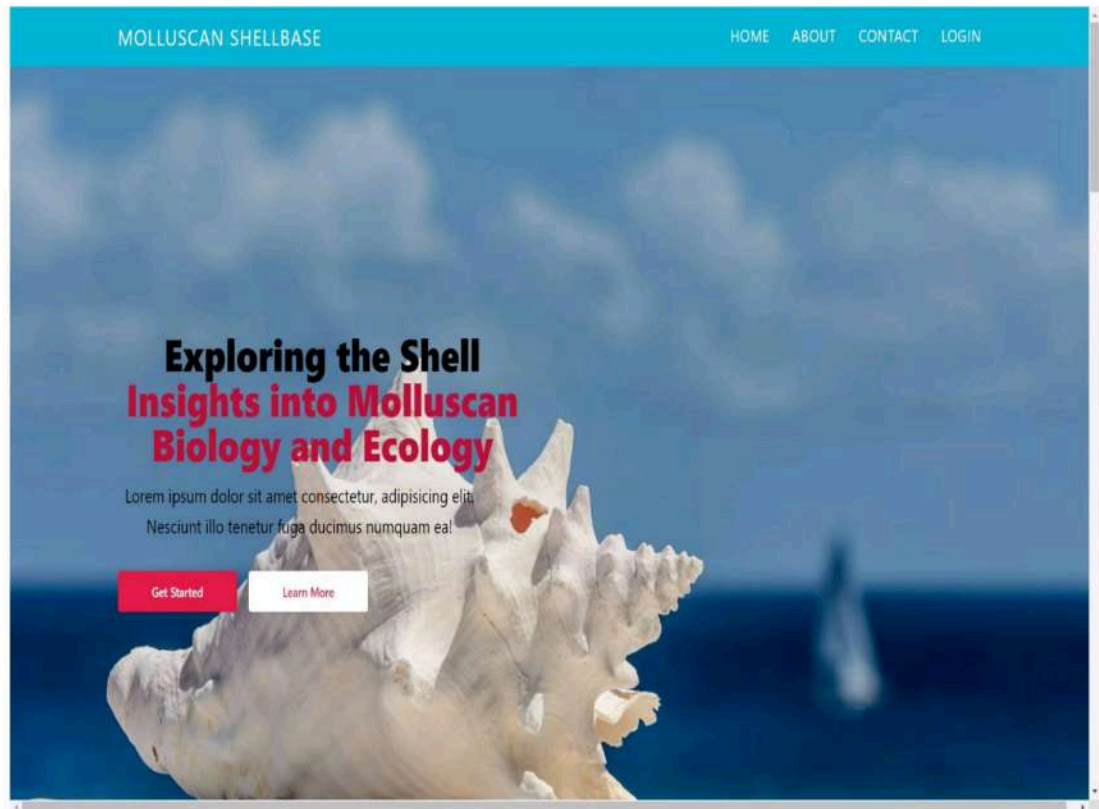
**Fig:5.2: Sign In**

## 5.2 Feasibility Study

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

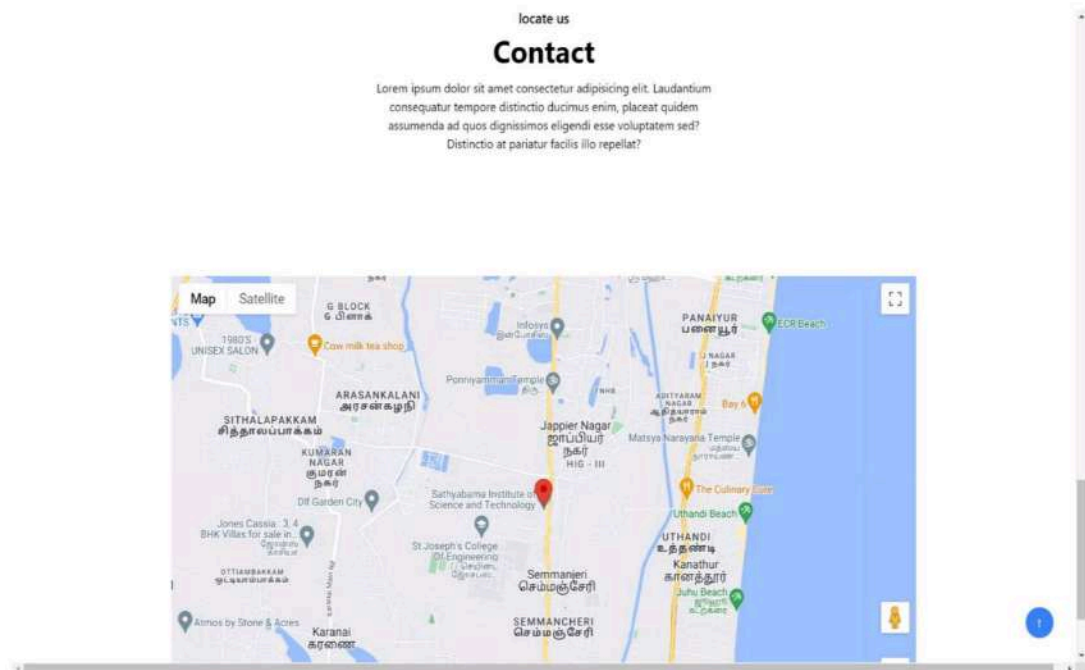
- ♦ Economic feasibility
- ♦ Technical feasibility
- ♦ Social feasibility



**Fig:5.3: Front Page**

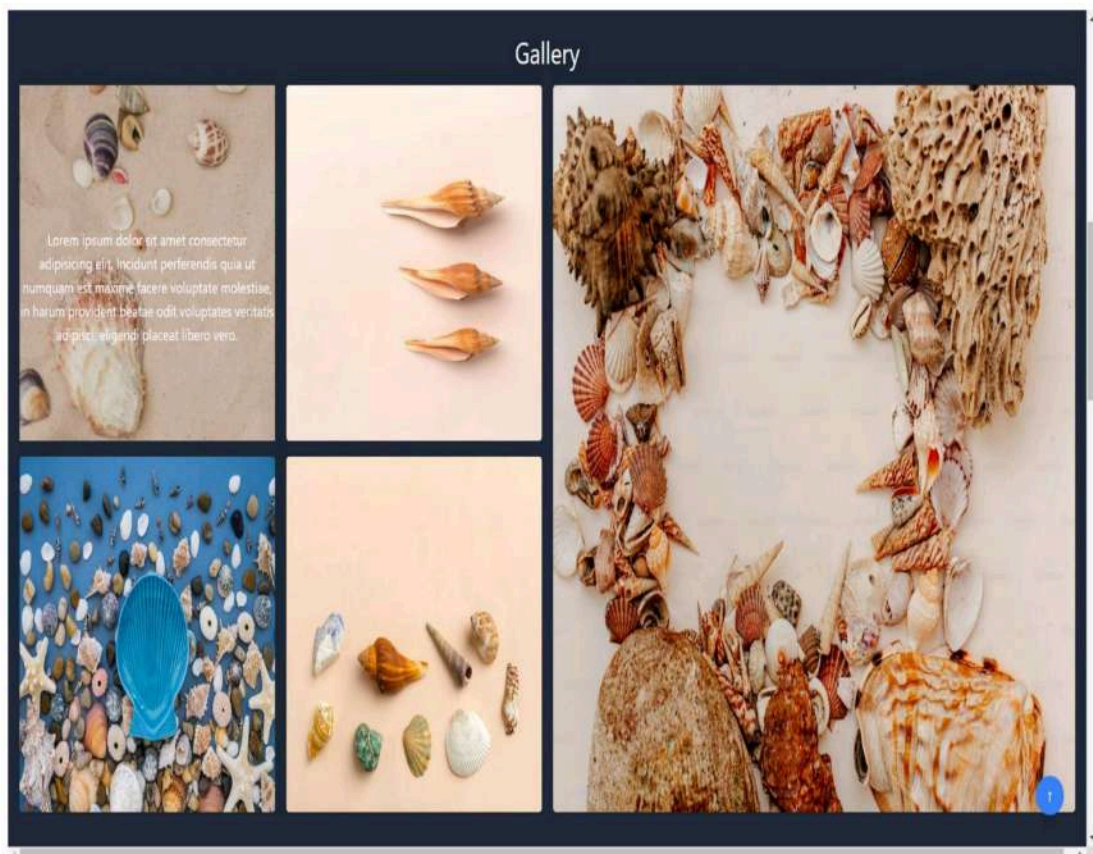
- ♦ **Economical Feasibility:** This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

- ♦ **Technical Feasibility:** This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.
- ♦ **Social Feasibility:** The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.



**Fig:5.4: Contact Page**

- ♦ **System Testing:** The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it.



**Fig5.5: Gallery of Molluscan Shells**

## **5.3 Types of Tests carried out**

### **5.3.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### **5.3.2 Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

- Test Results: All the test cases mentioned above passed successfully. No defects encountered.

- **Acceptance Testing:** User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

### **5.3.3 Functional testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

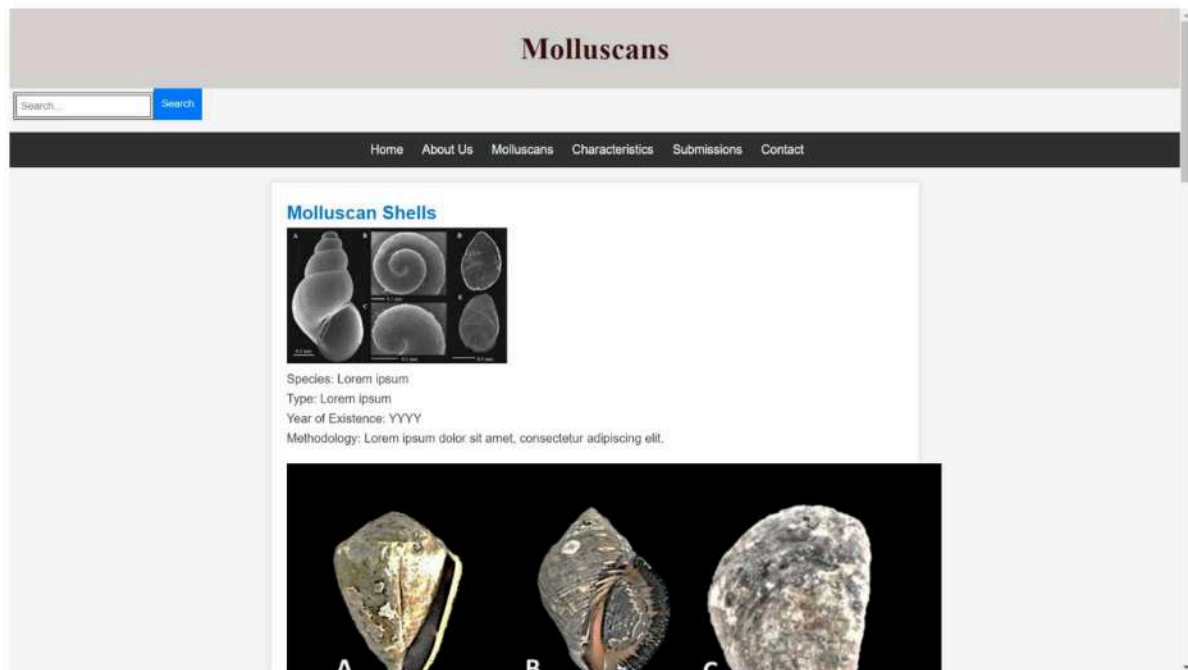
Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined. The test provides inputs and responds to outputs without considering how the software works.



**Fig:5.6: Sample Collection**

### 5.3.4 White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

### 5.3.5 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Test objectives**

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

**Features to be tested**

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

## **CHAPTER-6**

### **CONCLUSION**

In conclusion, the development of the Molluscan Shell Database represents a significant step towards enhancing research and conservation efforts in marine biology. By incorporating comprehensive functionalities for data entry, querying, visualization, and taxonomy, the system provides researchers with a powerful tool to explore and analyze molluscan shell information. The adherence to non-functional requirements such as performance, scalability, reliability, usability, and security ensures that the system is robust, efficient, and user-friendly. Ultimately, the database's ability to streamline data management, facilitate scientific inquiry, and contribute to the understanding of molluscan biodiversity underscores its importance in advancing marine science and conservation initiatives.

For future enhancement, integrating machine learning could automate data analysis, aiding in species identification and distribution prediction. Developing a mobile application would enable remote data collection and real-time contributions, enhancing field research efficiency. Collaborative features like forums and data-sharing options would foster community engagement and knowledge exchange among researchers. Semantic search and natural language processing could refine query accuracy and user experience, facilitating easier data retrieval. Integration of genomic data would deepen insights into molluscan evolution and diversity. Geospatial analysis tools would visualize habitat associations and environmental influences, aiding in spatially explicit research and conservation planning. Standardizing data formats and metadata schemas would enhance interoperability and facilitate data sharing across databases. Engaging with citizen scientists through outreach and citizen science projects would broaden data contributions and promote public involvement in marine science and conservation, enriching the database's scope and impact.

## REFERENCES

- [1] A. Y. Halevy, M. J. Franklin, and D. Maier. Principles of dataspace systems. In PODS, 2016.
- [2] Bouchet, P., Rocroi, J.P., Hausdorf, B., Kaim, A., Kano, Y., Nützel, A., & Strong, E.E. (2017). Revised classification, nomenclator and typification of gastropod and monoplacophoran families. *Malacologia*, 61(1-2), 1-526.
- [3] Brandt, R.A.M. (1974). The non-marine aquatic Mollusca of Thailand. *Archiv für Molluskenkunde*, 105(1-6), 1-423.
- [4] Cunha, C.M., & Absalão, R.S. (2003). *Moluscos marinhos do Brasil*. Museu Nacional.
- [5] Dayrat, B. (2012). Towards integrative taxonomy. *Biological Journal of the Linnean Society*, 107(1), 308-321.
- [6] Huber, M. (2010). *Compendium of bivalves*. A full-color guide to 3,300 of the world's marine bivalves: a status on bivalvia after 250 years of research.
- [7] P. Andritsos, R. J. Miller, and P. Tsaparas. Information-theoretic tools for mining database structure from large data sets. In SIGMOD, 2004.
- [8] Prudence Kadebu (2021) *International Journal of Modern Communication Technologies & Research (IJMCTR)* ISSN: 2321-0850, Volume-1, Issue-7, September 2013
- [9] R. McCann, A. Kramnik, W. Shen, V. Varadarajan, O. Sobulo, and A. Doan. Integrating data from disparate sources: A mass collaboration approach. In ICDE, 2005.
- [10] Rosenberg, G. (2014). A new critical estimate of named species-level diversity of the recent Mollusca. *American Malacological Bulletin*, 32(2), 308-322.
- [11] S. Sarawagi. Graphical models for structure extraction and information integration. Keynote talk and tutorial at ICDM-05, 2005.
- [12] Simone, L.R.L. (2016). *Land and freshwater mollusks of Brazil*. Museu de Zoologia da Universidade de São Paulo.
- [13] T. Johnson and T. Dasu. Data quality and data cleaning: An overview. In SIGMOD, 2003. [8] D. R. Karger, K. Bakshi, D. Huynh, D. Quan, and V. Sinha. Haystack: A general-purpose information management tool for end users based on

semistructured data. In CIDR, 2005.

[14] Taylor, J.D., Kennedy, W.J., & Hall, A. (1993). The shell structure and mineralogy of the bivalvia. Oxford University Press.

[15] Turgeon, D.D., Quinn, J.F., Bogan, A.E., Coan, E.V., Hochberg, F.G., Lyons, W.G., & Rosenberg, G. (1998). Common and scientific names of aquatic invertebrates from the United States and Canada: Mollusks. American Fisheries Society Special Publication, 26.

[16] W. Cohen. Information extraction. Tutorial, [www.cs.cmu.edu/wcohen/ie-survey.ppt](http://www.cs.cmu.edu/wcohen/ie-survey.ppt), 2013. [4] A. Doan, R. Ramakrishnan, F. Chen, P. DeRose, Y. Lee, R. McCann, M. Sayyadian, and W. Shen. Community information management. In IEEE Data Engineering Bulletin, Special Issue on Probabilistic Databases, volume 29, 2006.

[17] Williams, S.T., & Reid, D.G. (2014). Speciation and diversity on tropical rocky shores: a global phylogeny of snails of the genus Echinolittorina. *Evolutionary Ecology Research*, 6(3), 391-420.

[18] Y. Cai, X. L. Dong, A. Y. Halevy, J. M. Liu, and J. Madhavan. Personal information management with semex. In SIGMOD, 2005.

## **APPENDIX A**

### **STANDARDS**

[1] Data Standards: Adherence to standardized data formats, metadata schemas, and vocabularies, such as Darwin Core for biodiversity data and Dublin Core for metadata, ensures interoperability and compatibility with external databases and repositories.

[2] Web Development Standards: Compliance with web development standards and best practices, such as HTML5, CSS3, and ECMAScript 6, ensures cross-browser compatibility, accessibility, and usability of the platform across different devices and platforms.

[3] Security Standards: Implementation of industry-standard security protocols and encryption algorithms, such as HTTPS, JWT authentication, and OAuth for user authentication and authorization, ensures the confidentiality, integrity, and availability of research data and user information.

[4] Accessibility Standards: Compliance with web accessibility standards, such as WCAG (Web Content Accessibility Guidelines), ensures that the platform is accessible to users with disabilities, including those using assistive technologies such as screen readers and keyboard navigation.

## **APPENDIX B:**

### **CONSTRAINTS**

[1] Resource Constraints: Limited resources, including time, budget, and human resources, impose constraints on the scope, complexity, and timeframe of the project. Balancing project requirements with resource limitations requires careful prioritization, planning, and resource allocation.

[2] Technical Constraints: Technical limitations, such as hardware capabilities, software dependencies, and compatibility issues, may constrain the functionality, performance, and scalability of the platform. Addressing technical constraints requires innovative problem-solving and adaptation of technology solutions to meet project objectives.

[3] Regulatory Constraints: Regulatory requirements, such as data privacy laws (e.g., GDPR) and intellectual property rights, impose constraints on data handling, storage, and sharing practices. Compliance with regulatory constraints ensures legal and ethical integrity in the management and dissemination of research data.

[4] User Constraints: User preferences, expectations, and limitations may constrain the design, usability, and functionality of the platform. Conducting user research, feedback sessions, and usability testing helps identify and address user constraints to enhance user satisfaction and engagement.

## **APPENDIX C:**

### **TRADEOFFS**

[1] Performance vs. Scalability: Tradeoffs between performance and scalability may arise when optimizing the platform for efficient data processing and storage while accommodating increasing user traffic and data volumes. Balancing performance optimizations with scalability enhancements requires careful architectural design and performance testing.

[2] Functionality vs. Usability: Tradeoffs between functionality and usability may occur when adding new features and functionalities to the platform while maintaining a user-friendly and intuitive interface. Prioritizing essential features and minimizing complexity ensures a balance between functionality and usability to meet user needs effectively.

[3] Security vs. Accessibility: Tradeoffs between security and accessibility may arise when implementing security measures, such as CAPTCHA verification and multi-factor authentication, that may hinder accessibility for users with disabilities. Balancing security requirements with accessibility standards ensures both data protection and inclusivity for all users.

[4] Cost vs. Quality: Tradeoffs between cost and quality may arise when selecting technologies, resources, and services for the project. Investing in high-quality tools and services may incur higher costs but ensures reliability, performance, and maintainability in the long run.